Falcon 9 – Data Science Project

Martin R.

21.3.2022

Executive Summary

- The goal of this presentation is to display the prediction of landing of Falcon 9 based on data-driven insights.
- The findings were obtained through:

    1. Webscraping data collection
    2. Data Wrangling
    3. EDA Analysis with SQL
    4. Matplotlib and Folium
    5. Machine Learning and Predictive Analysis

Table of Content

- Introduction
- Methodology
- Results - Webscraping
  - Exploring and Preparing Data
  - SQL Analysis
  - Folium Visualization
  - Machine Learning Prediction
- Conclusion
- Discussion
- Appendix

Introduction

About SpaceX

- SpaceX is one of the most successful spacecraft company owned by Elon Musk. This company has many accomplishments such as sending Starlink internet satellite to the space or sending spacecraft to the International Space Station. One of the reason why SpaceX is so successful is because their rocket launches are not much expensive. That is because their Falcon 9 can reuse the stage.

Goal of this project

- The goal of this project is to determine the future of rocket launches using data science.

Methodology

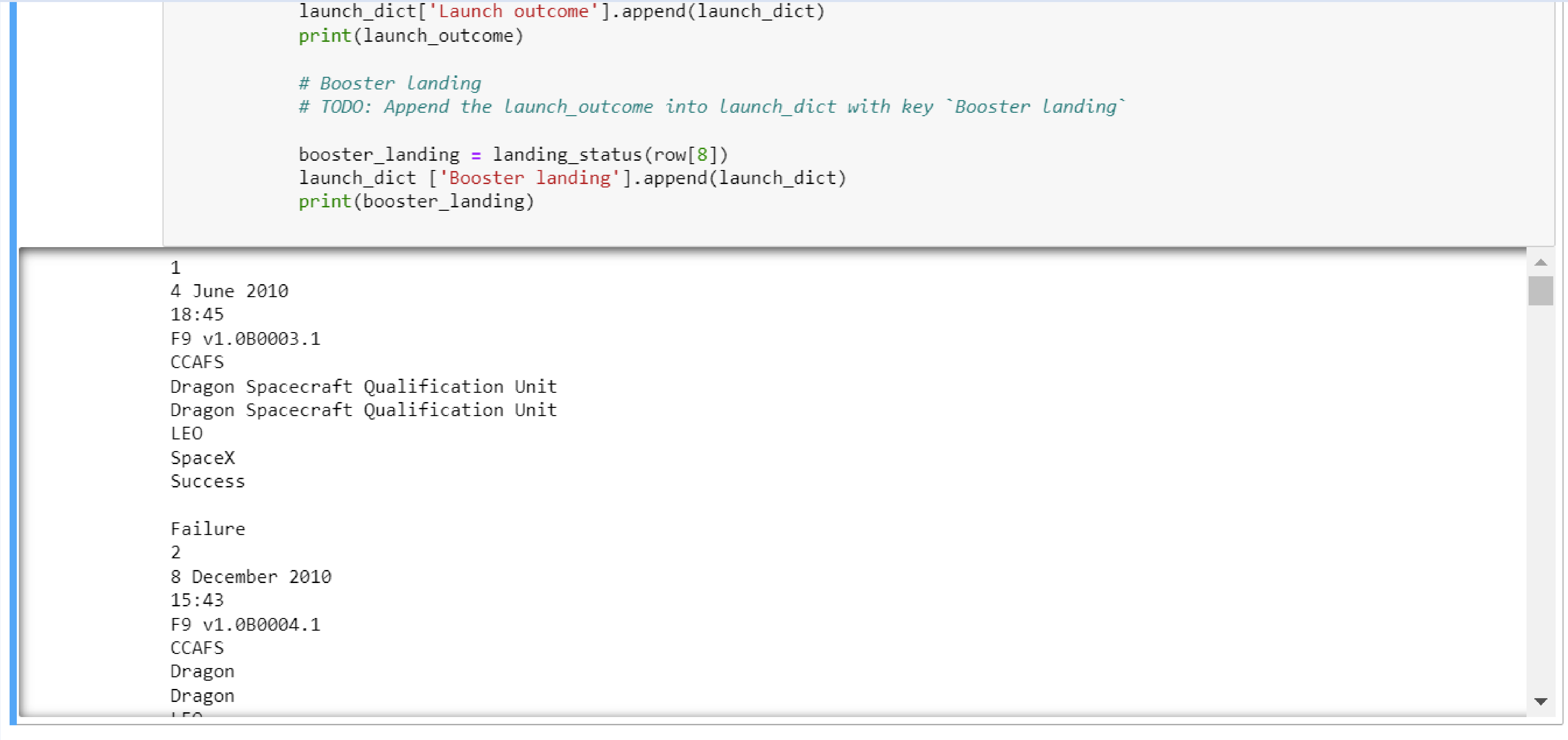
- **Data Collection**
- Data collection is how we gather data in this project these tools were used:
- API
- Webscraping – BeautifulSoup
- Data Wrangling - Pandas
- SQL analysis

- **Data Preparation**
- Data preparation is how we cleanse and prepare data in this project these tools were used:
- Numpy
- Matplotlib
- Seaborn
- Machine Learning

- **Data Modeling**
- Data modeling is presenting data in this project these tools were used:
- Folium
- Plotly

Results

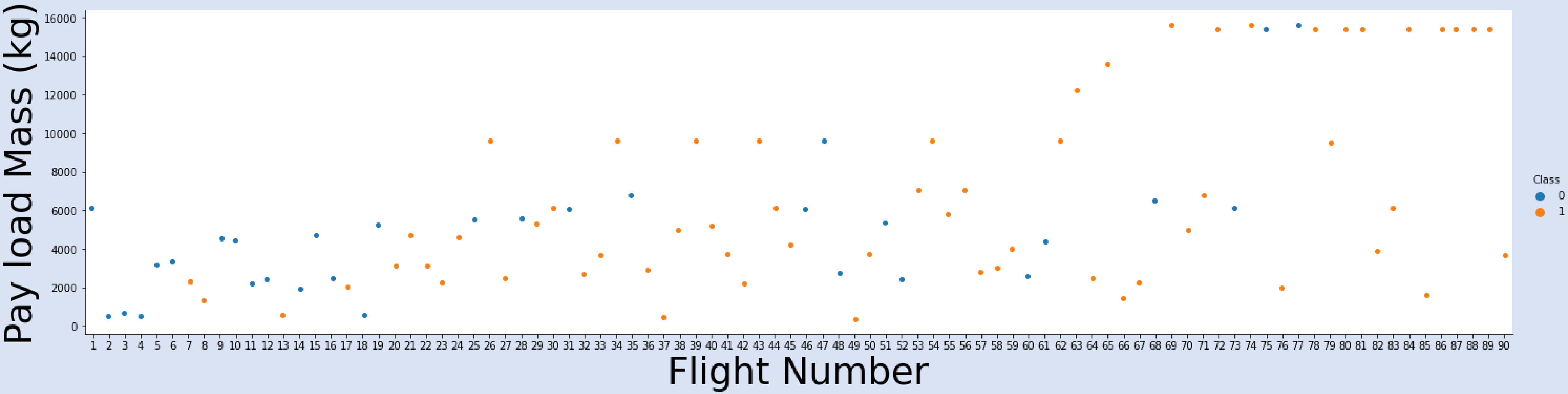
Webscraping

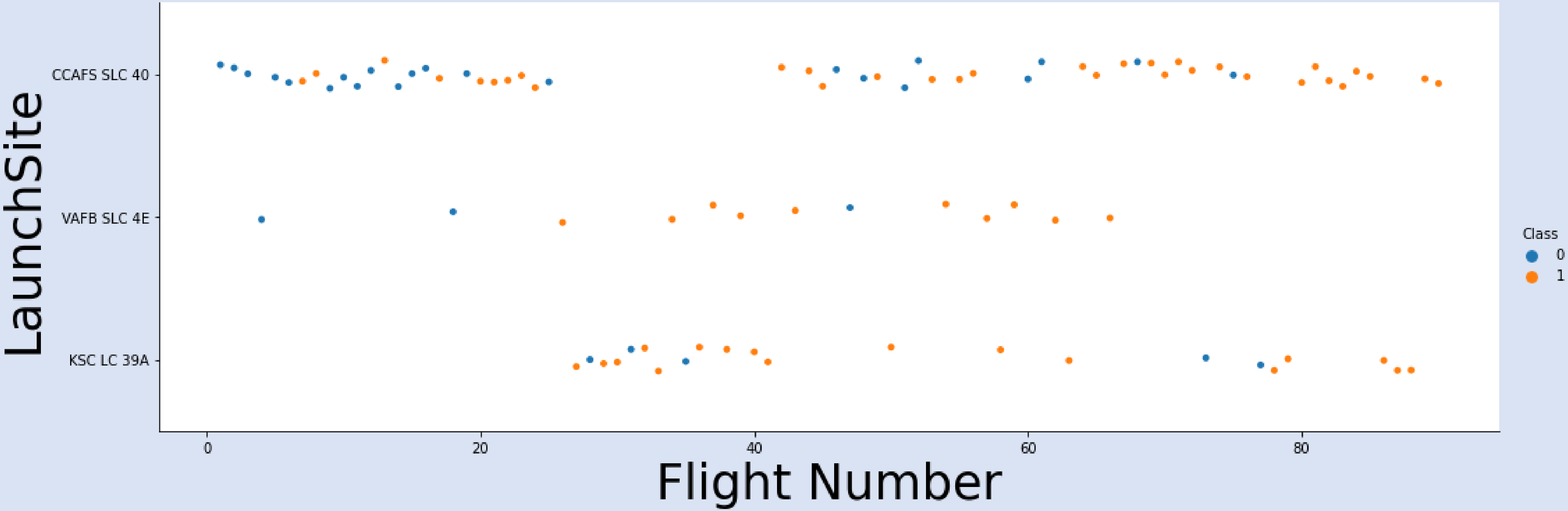


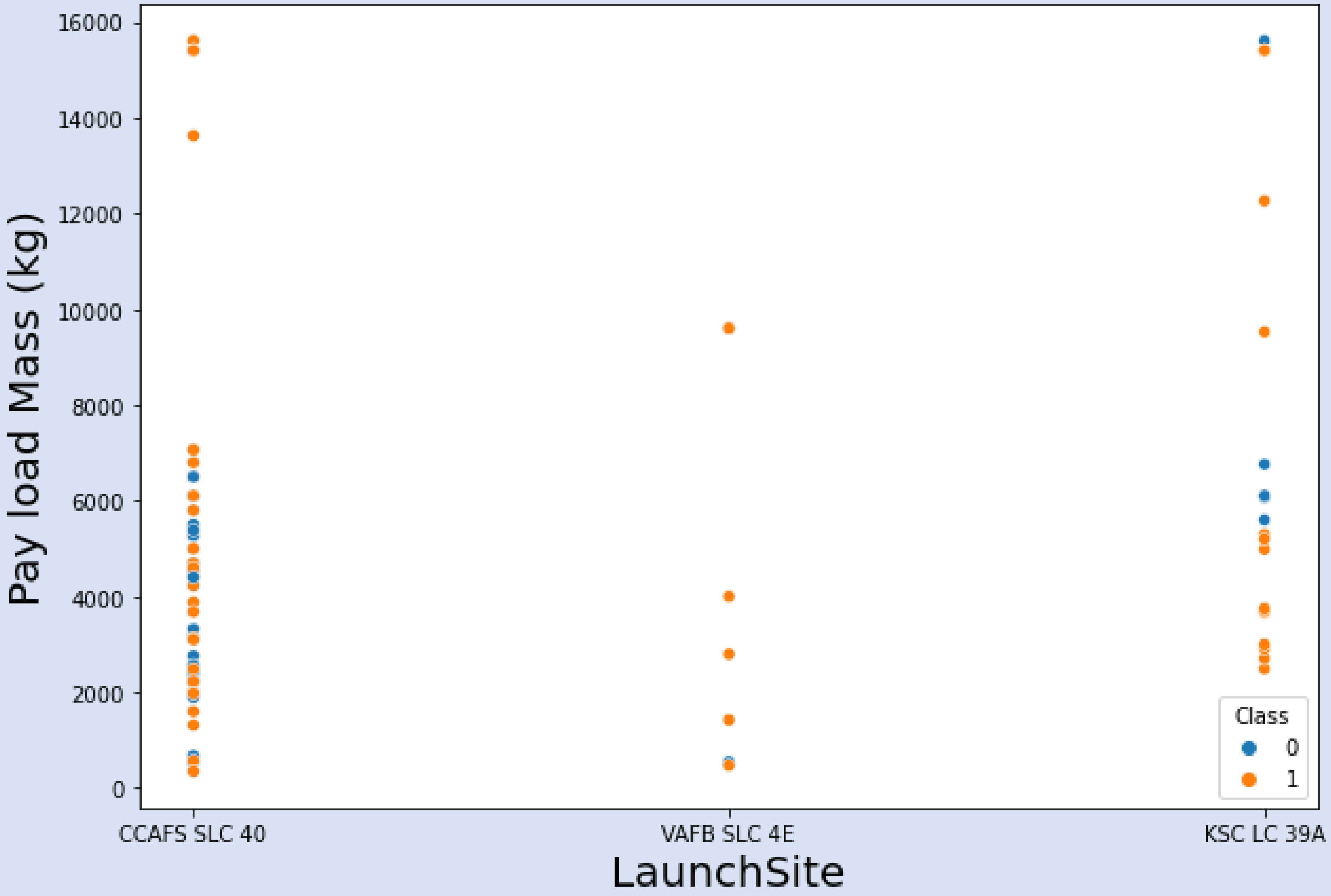
```
        launch_dict['Launch outcome'].append(launch_dict)
        print(launch_outcome)

        # Booster landing
        # TODO: Append the launch_outcome into launch_dict with key `Booster landing`

        booster_landing = landing_status(row[8])
        launch_dict ['Booster landing'].append(launch_dict)
        print(booster_landing)
```

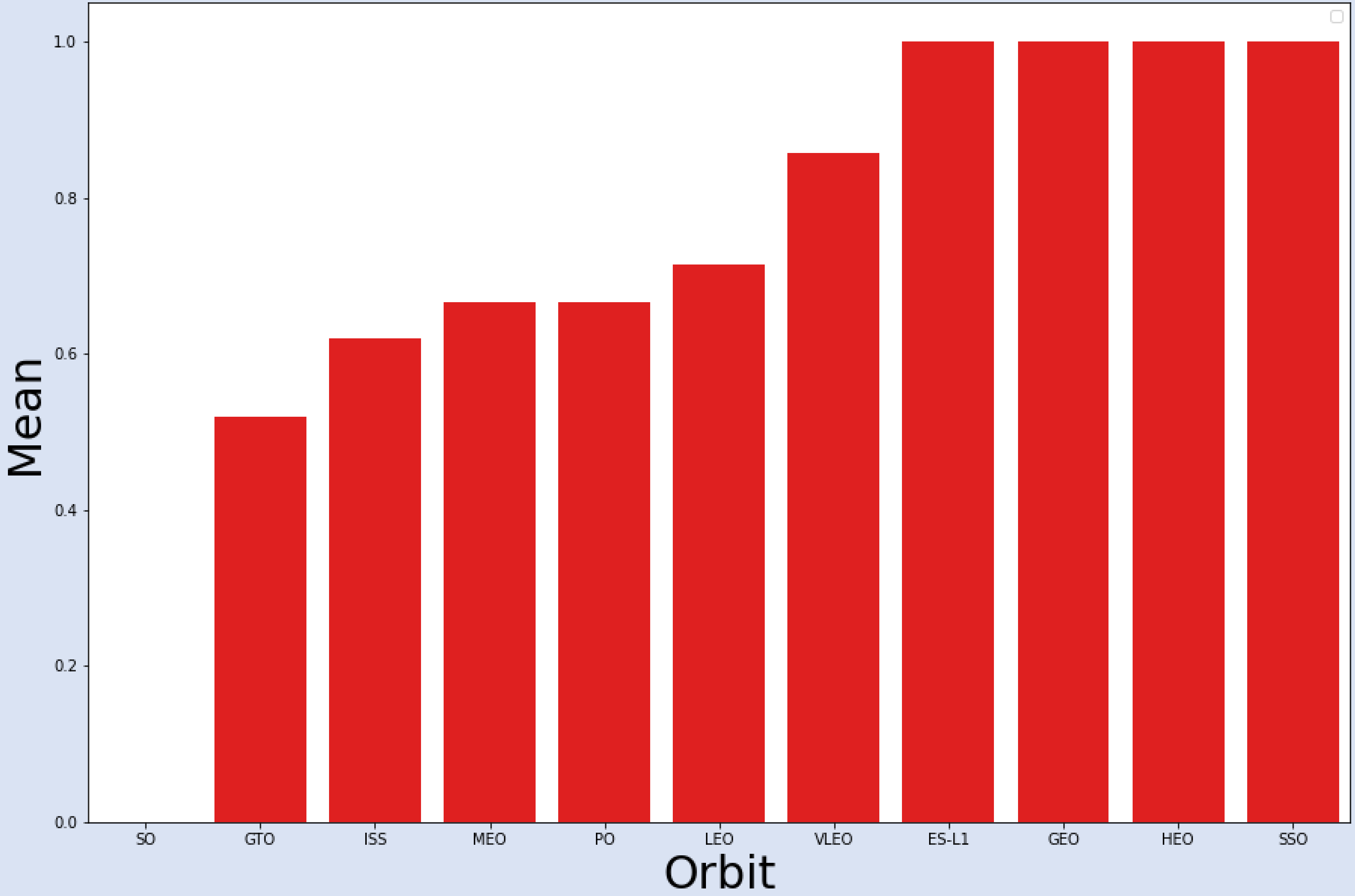
```
1
4 June 2010
18:45
F9 v1.0B0003.1
CCAFS
Dragon Spacecraft Qualification Unit
Dragon Spacecraft Qualification Unit
LEO
SpaceX
Success

Failure
2
8 December 2010
15:43
F9 v1.0B0004.1
CCAFS
Dragon
Dragon
LEO
```

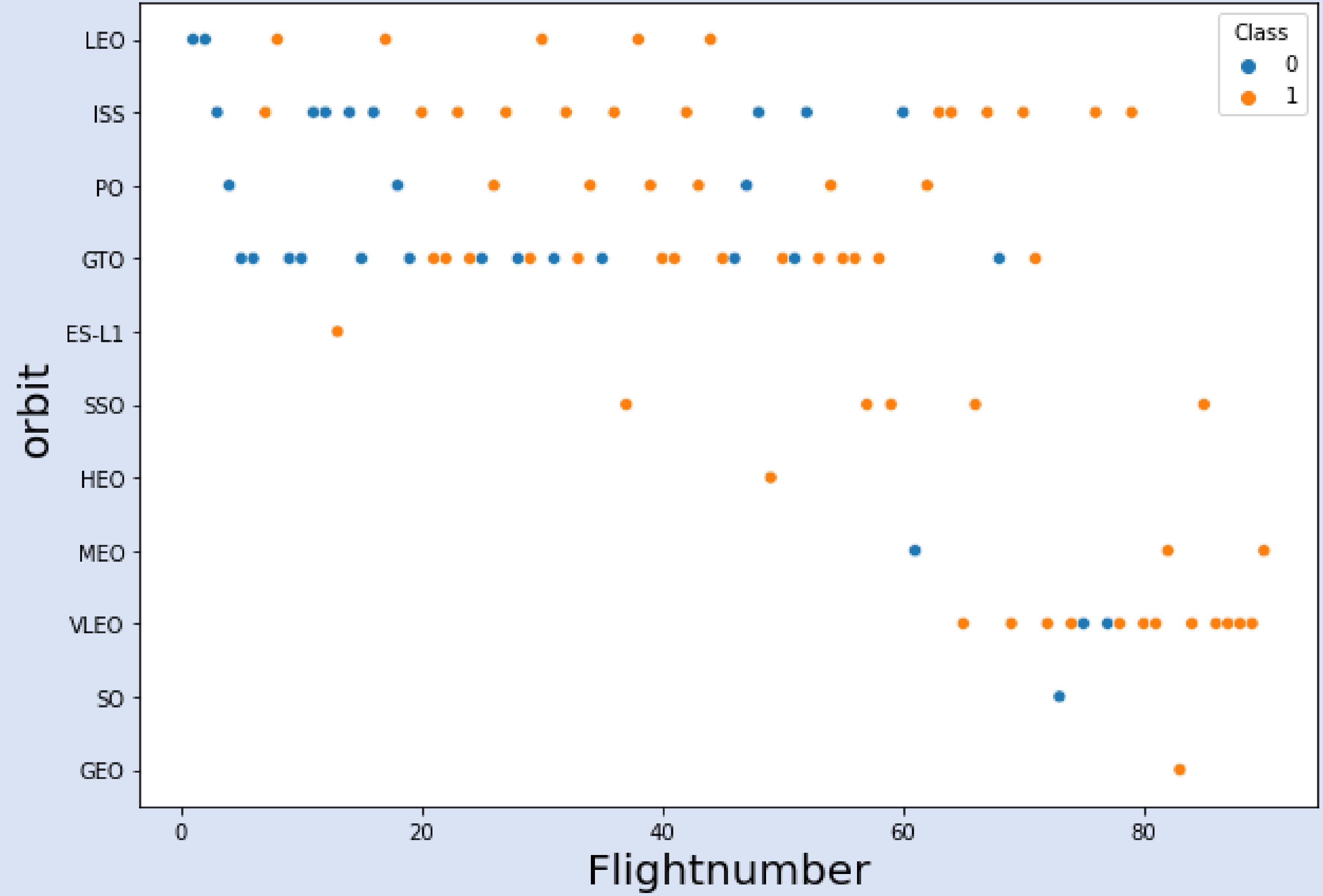
Exploring and Preparing Data

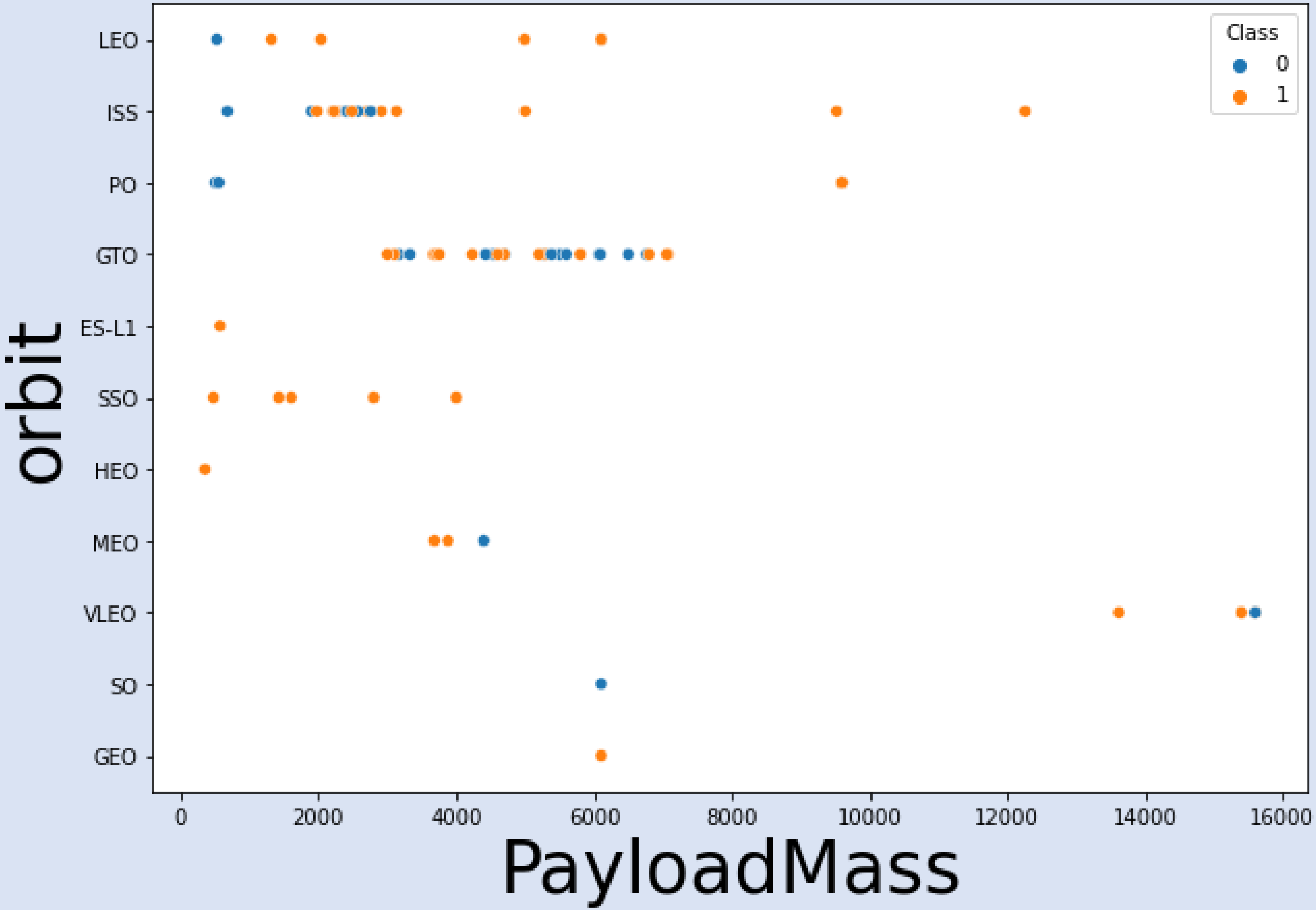

Pay load Mass (kg)
0
2000
4000
6000
8000
10000
12000
14000
16000
Flight Number
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
Class
0
1


LaunchSite
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
0
20
40
60
80
Flight Number
Class
0
1

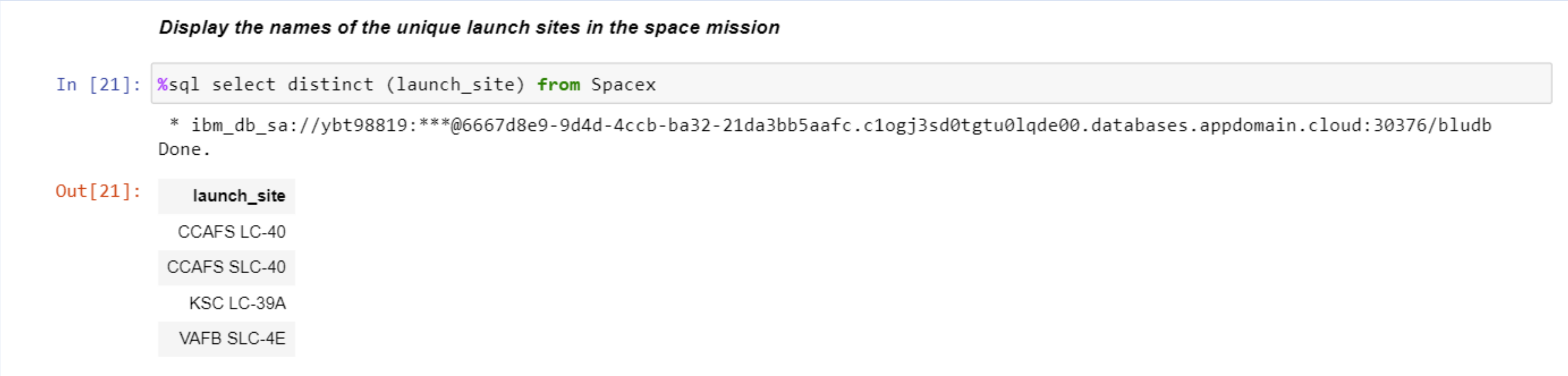

Pay load Mass (kg)
16000
14000
12000
10000
8000
6000
4000
2000
0
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
LaunchSite
Class
● 0
● 1


Mean
Orbit
1.0
0.8
0.6
0.4
0.2
0.0
SO
GTO
ISS
MEO
PO
LEO
VLEO
ES-L1
GEO
HEO
SSO


orbit
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
0
20
40
60
80
Flightnumber
Class
● 0
● 1


Class
● 0
● 1
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
orbit
0
2000
4000
6000
8000
10000
12000
14000
16000
PayloadMass

SQL analysis



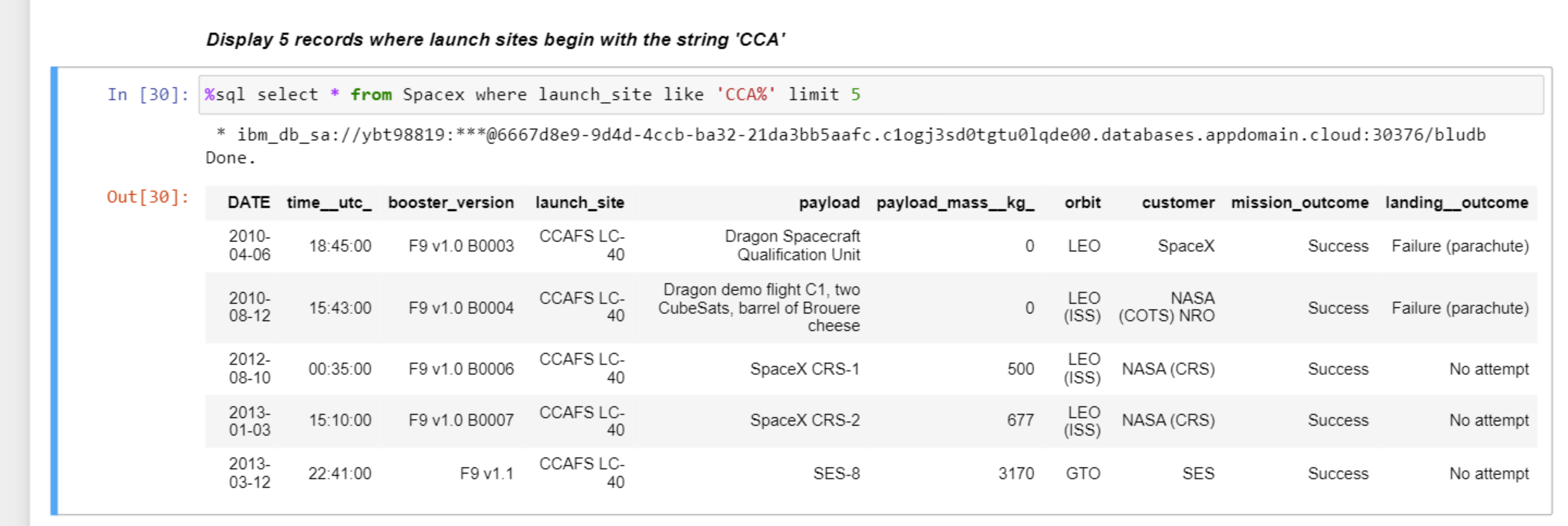
Display the names of the unique launch sites in the space mission

```
In [21]: %sql select distinct (launch_site) from Spacex
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[21]:

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E



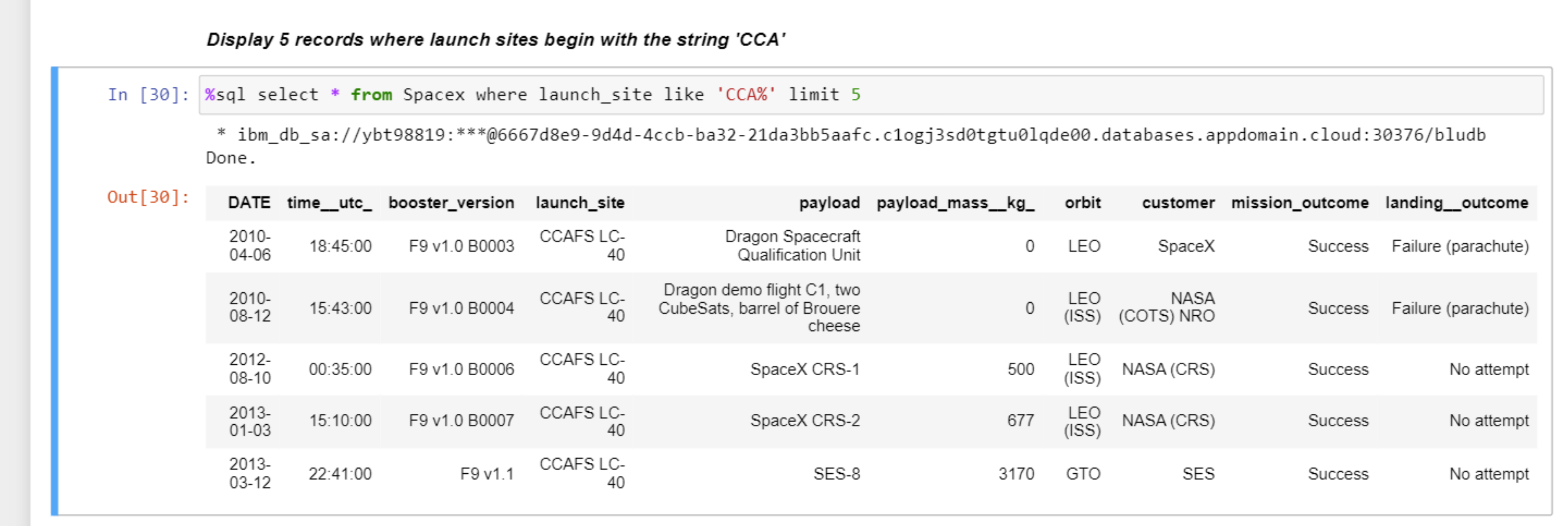
Display 5 records where launch sites begin with the string 'CCA'

```
In [30]: %sql select * from Spacex where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[30]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt



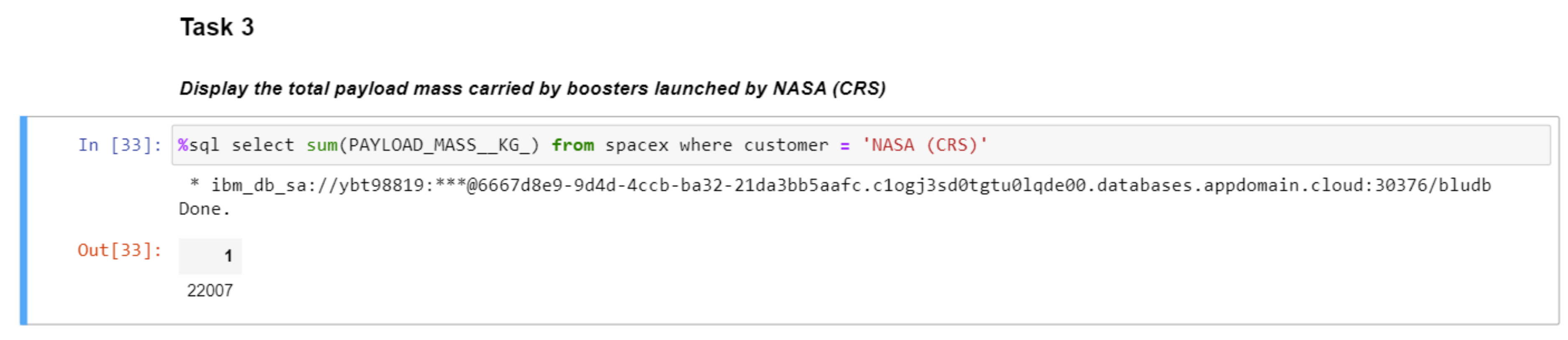
Display 5 records where launch sites begin with the string 'CCA'

```
In [30]: %sql select * from Spacex where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[30]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt



Task 3

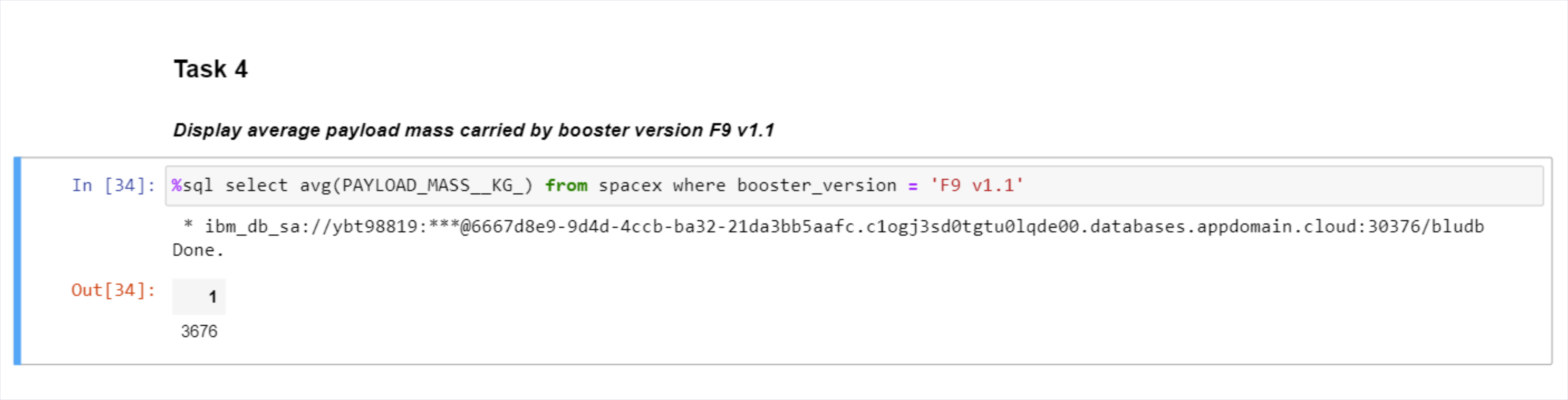
Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [33]: %sql select sum(PAYLOAD_MASS__KG_) from spacex where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[33]:

1
22007



Task 4

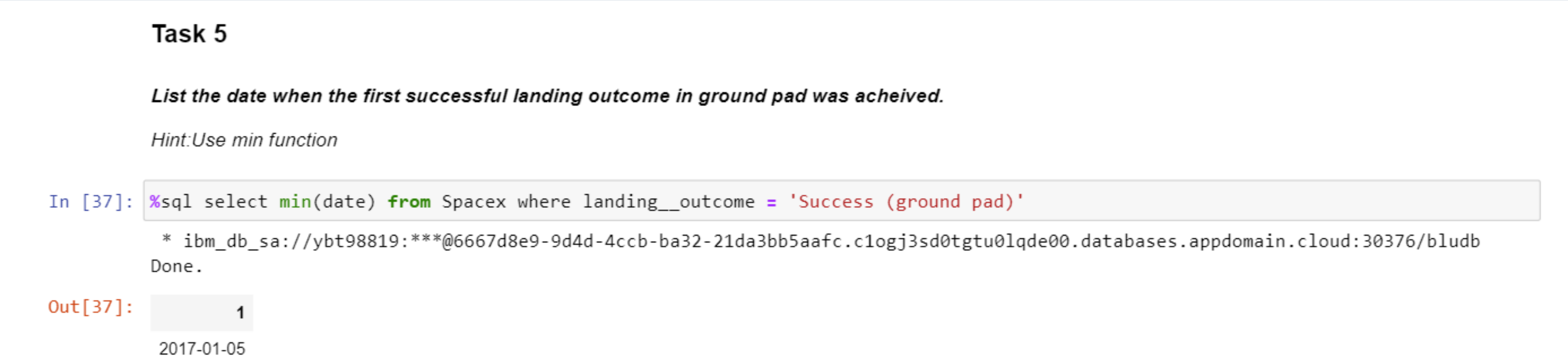
Display average payload mass carried by booster version F9 v1.1

```
In [34]: %sql select avg(PAYLOAD_MASS__KG_) from spacex where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[34]:

1
3676



Task 5

List the date when the first successful landing outcome in ground pad was acheived.

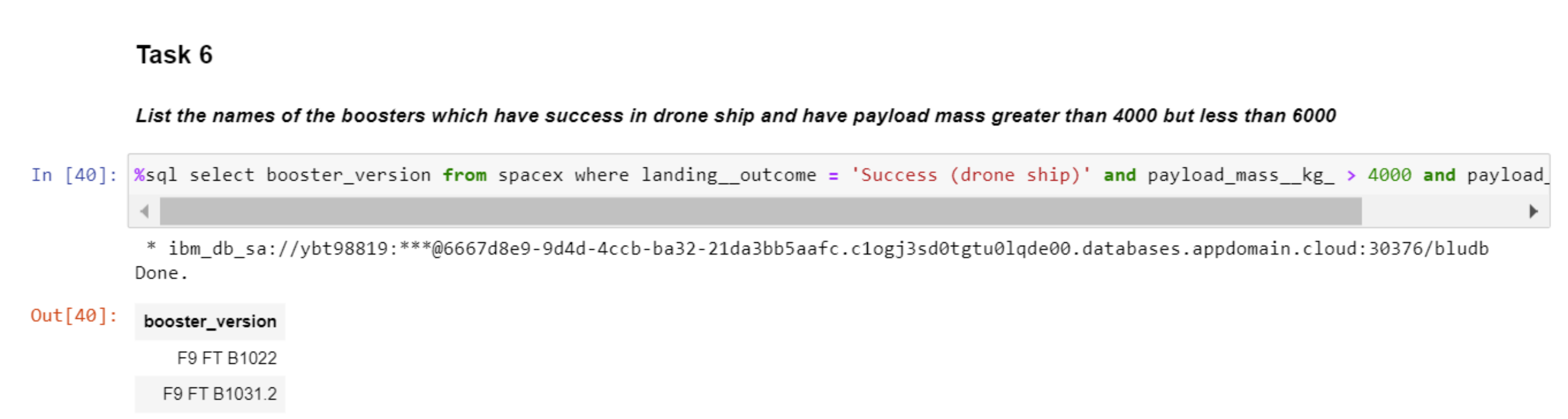
Hint:Use min function

```
In [37]: %sql select min(date) from Spacex where landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[37]:

1
2017-01-05



Task 6

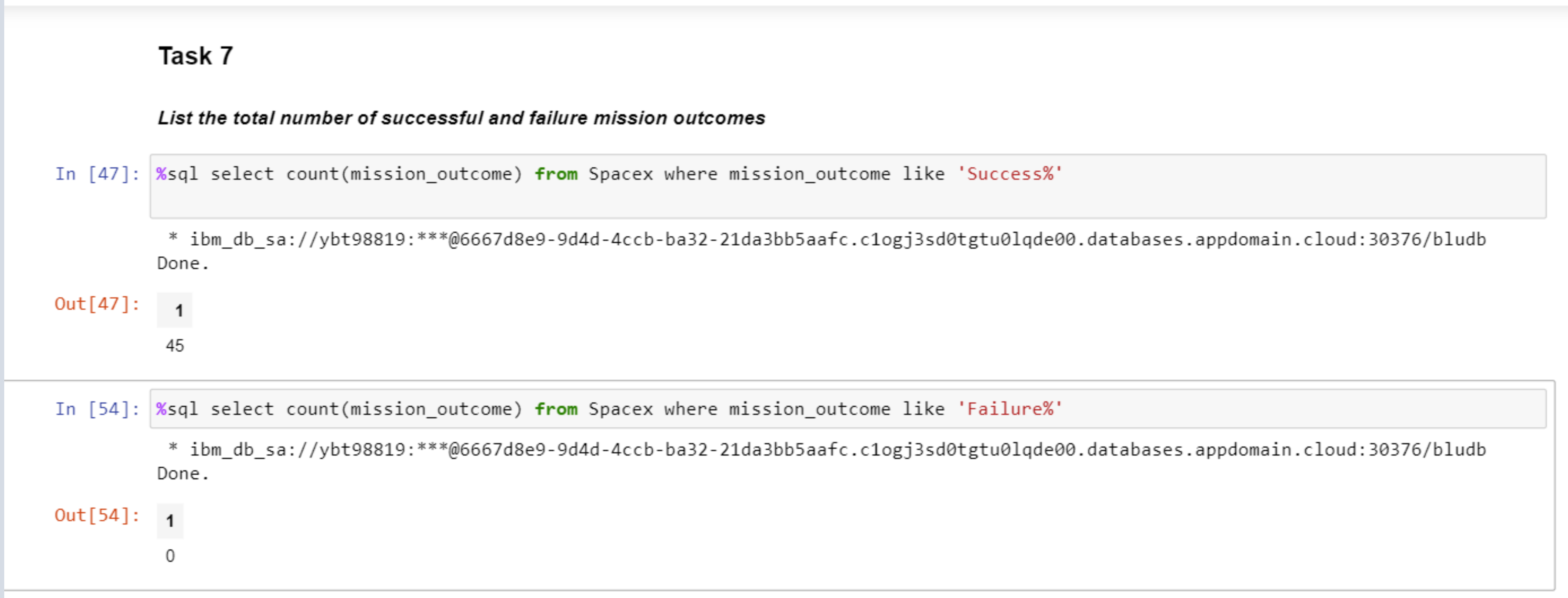
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
In [40]: %sql select booster_version from spacex where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ > 4000 and payload_
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[40]:

booster_version
F9 FT B1022
F9 FT B1031.2



Task 7

List the total number of successful and failure mission outcomes

```
In [47]: %sql select count(mission_outcome) from Spacex where mission_outcome like 'Success%'
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[47]:

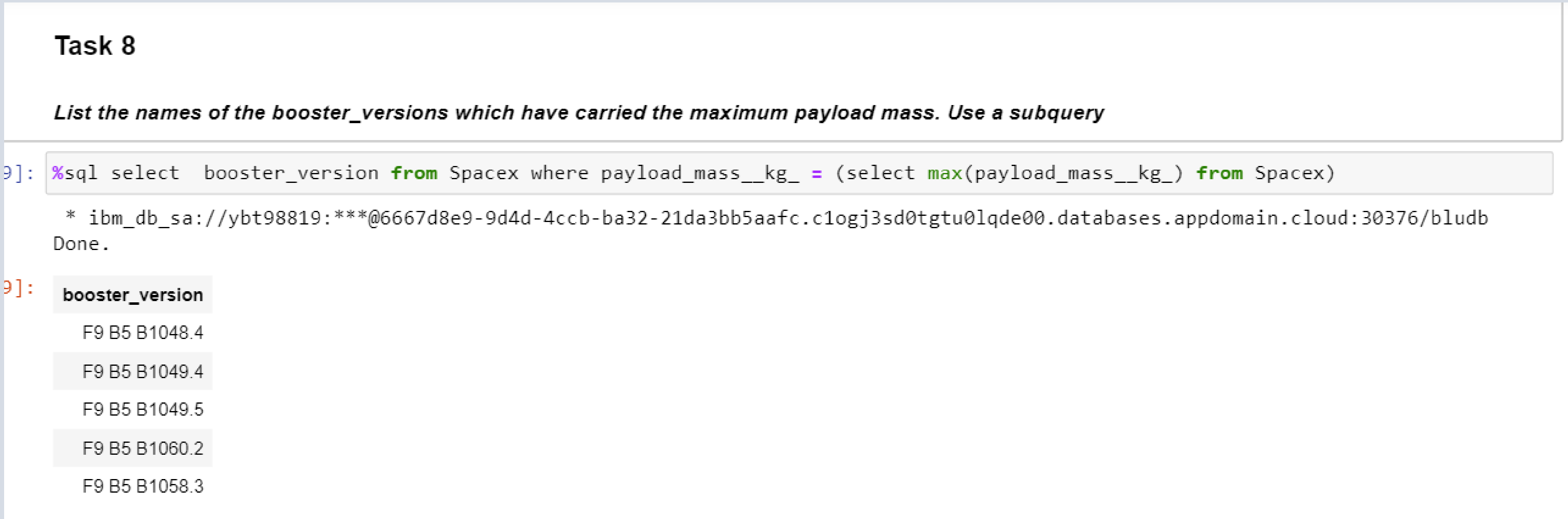
1
45

```
In [54]: %sql select count(mission_outcome) from Spacex where mission_outcome like 'Failure%'
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[54]:

1
0



Task 8

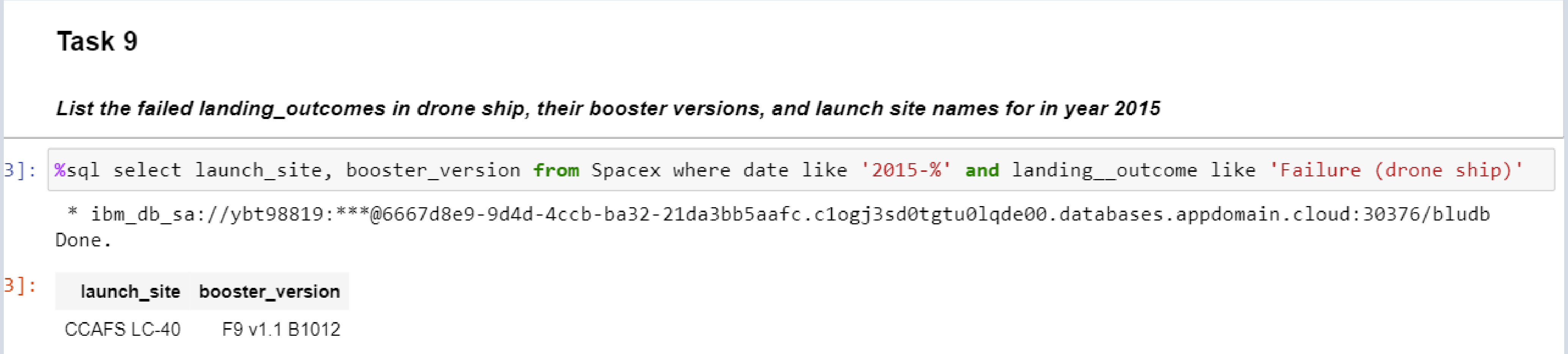
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
9]: %sql select  booster_version from Spacex where payload_mass__kg_ = (select max(payload_mass__kg_) from Spacex)
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

9]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3



Task 9

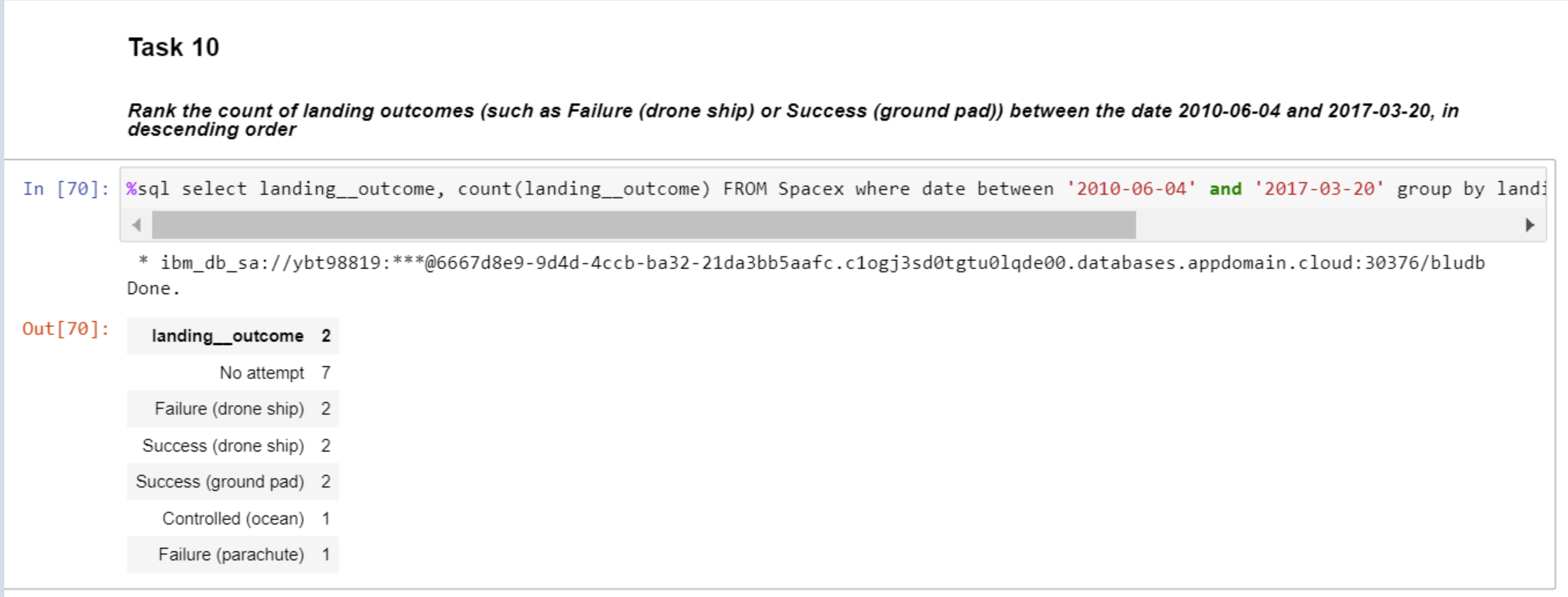
List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
3]: %sql select launch_site, booster_version from Spacex where date like '2015-%' and landing__outcome like 'Failure (drone ship)'
```

```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

3]:

launch_site	booster_version
CCAFS LC-40	F9 v1.1 B1012



Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

In [70]:

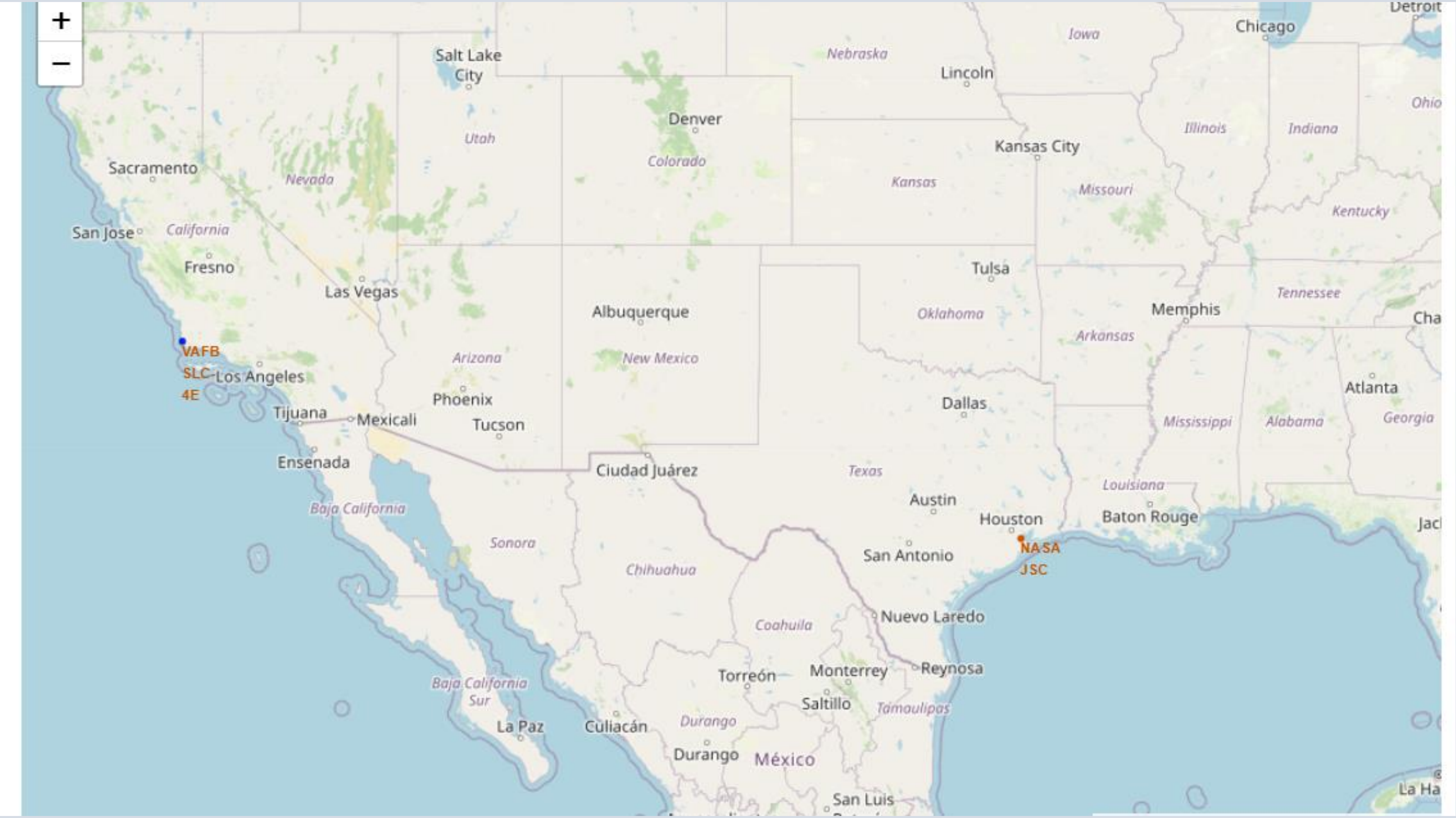
```
%sql select landing__outcome, count(landing__outcome) FROM Spacex where date between '2010-06-04' and '2017-03-20' group by landi
```

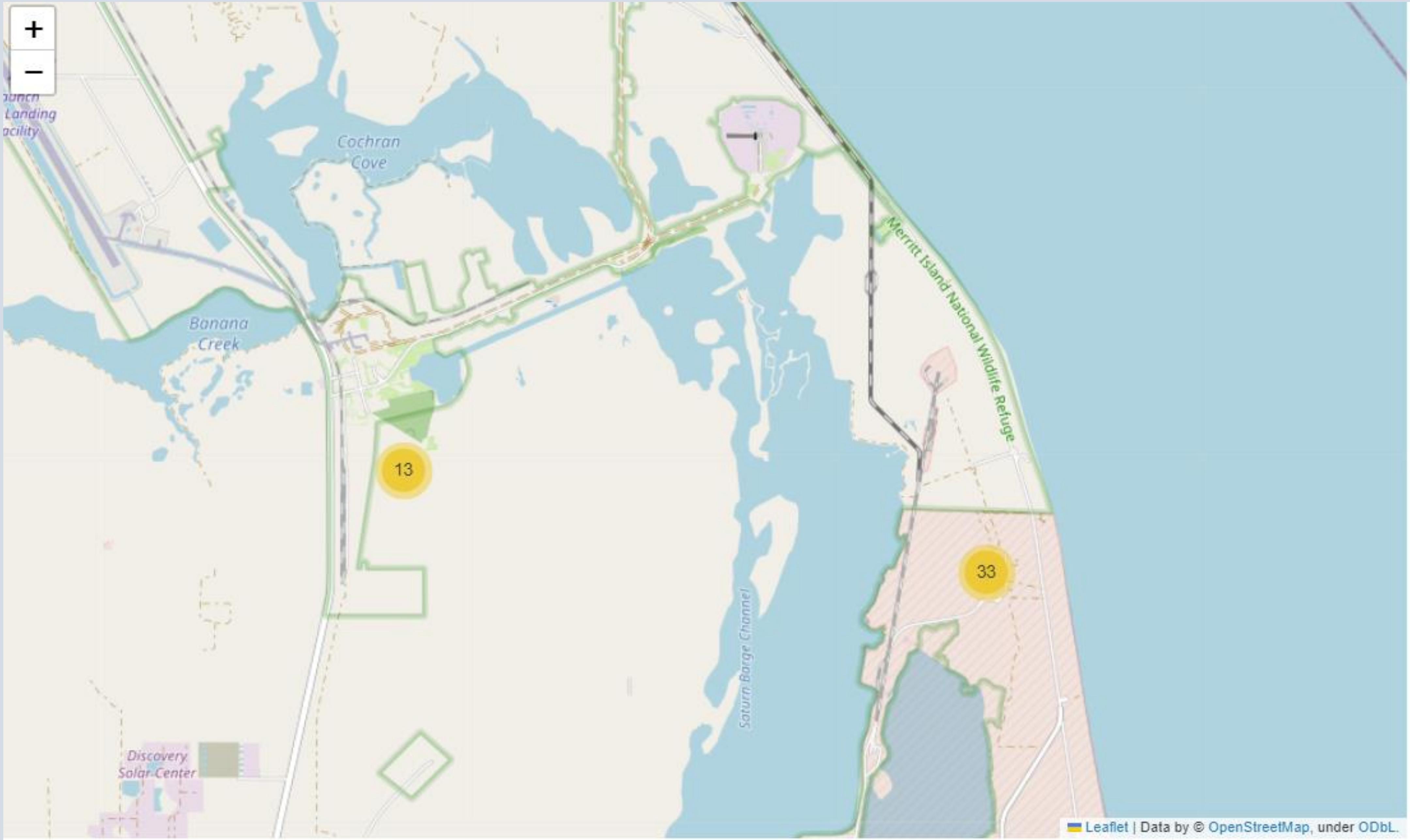
```
 * ibm_db_sa://ybt98819:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[70]:

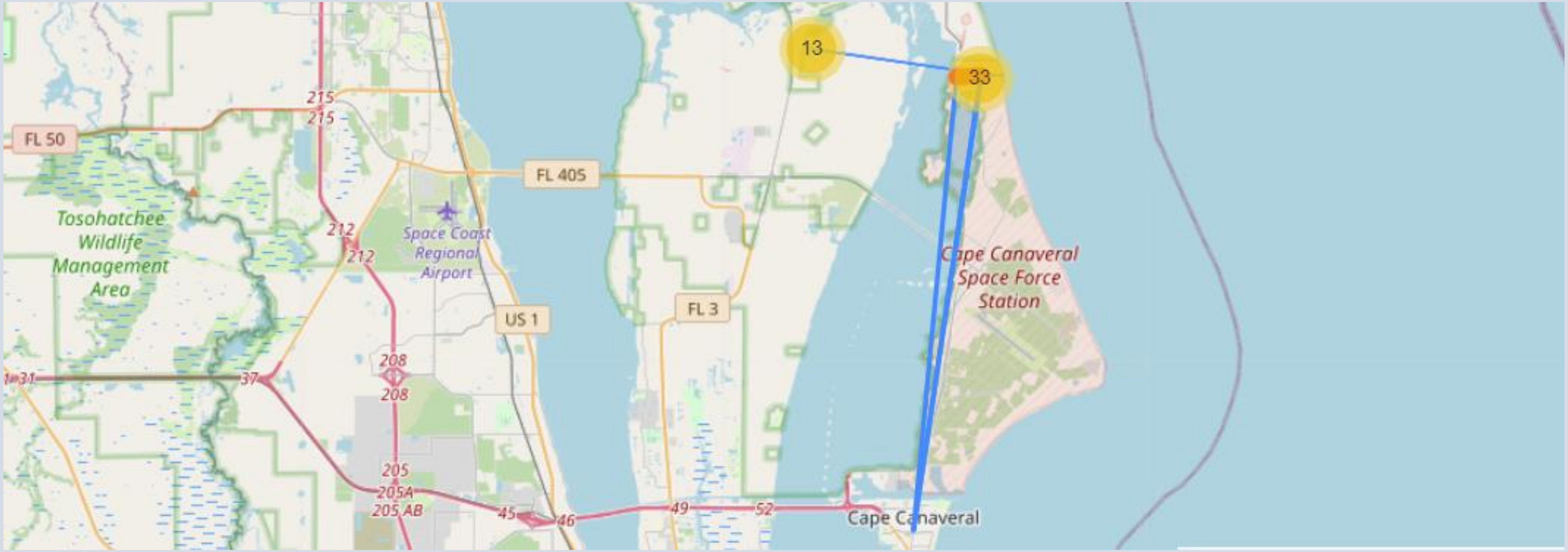
landing__outcome	2
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

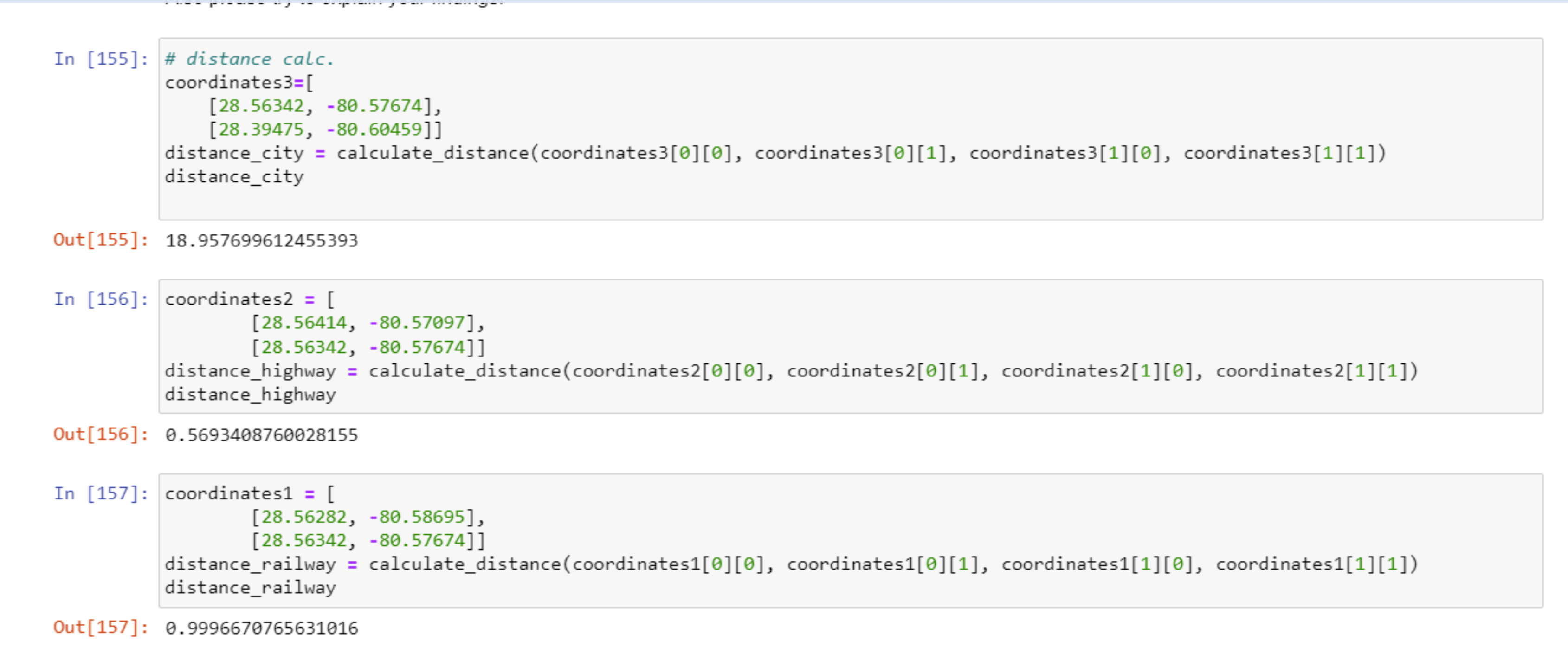
Folium visualization


+
−
VAFB
SLC-4E
NASA
JSC
Salt Lake City
Denver
Lincoln
Chicago
Detroit
Iowa
Nebraska
Utah
Colorado
Kansas City
Illinois
Indiana
Ohio
Sacramento
Nevada
Kansas
Missouri
Kentucky
San Jose
California
Fresno
Las Vegas
Tulsa
Tennessee
Memphis
Albuquerque
Oklahoma
Arkansas
Cha
Arizona
New Mexico
Los Angeles
Atlanta
Phoenix
Dallas
Tijuana
Mexicali
Tucson
Mississippi
Alabama
Georgia
Ensenada
Ciudad Juárez
Texas
Louisiana
Baja California
Austin
Houston
Baton Rouge
Jac
Sonora
San Antonio
Chihuahua
Nuevo Laredo
Coahuila
Torreón
Monterrey
Reynosa
Baja California Sur
Saltillo
Tamaulipas
La Paz
Culiacán
Durango
Durango
México
San Luis
La Ha



🇺🇦 Leaflet | Data by © OpenStreetMap, under ODbL.


13
33
FL 50
FL 405
215
215
212
212
Tosohatchee
Wildlife
Management
Area
Space Coast
Regional
Airport
US 1
FL 3
Cape Canaveral
Space Force
Station
31
37
208
208
205
205A
205 AB
45
46
49
52
Cape Canaveral



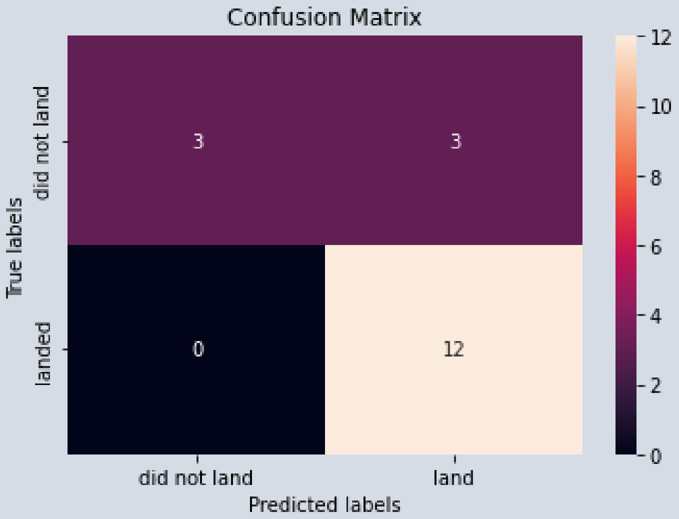
```
In [155]: # distance calc.
          coordinates3=[
              [28.56342, -80.57674],
              [28.39475, -80.60459]]
          distance_city = calculate_distance(coordinates3[0][0], coordinates3[0][1], coordinates3[1][0], coordinates3[1][1])
          distance_city

Out[155]: 18.957699612455393
```

```
In [156]: coordinates2 = [
                  [28.56414, -80.57097],
                  [28.56342, -80.57674]]
          distance_highway = calculate_distance(coordinates2[0][0], coordinates2[0][1], coordinates2[1][0], coordinates2[1][1])
          distance_highway

Out[156]: 0.5693408760028155
```

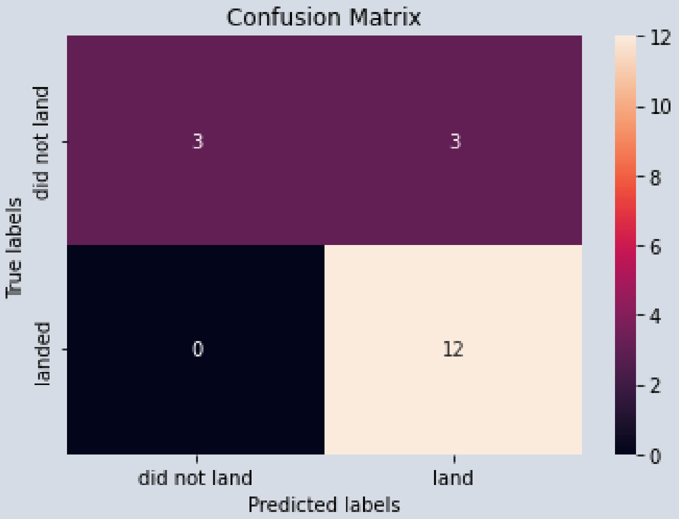
```
In [157]: coordinates1 = [
                  [28.56282, -80.58695],
                  [28.56342, -80.57674]]
          distance_railway = calculate_distance(coordinates1[0][0], coordinates1[0][1], coordinates1[1][0], coordinates1[1][1])
          distance_railway

Out[157]: 0.9996670765631016
```

Machine Learning Prediction

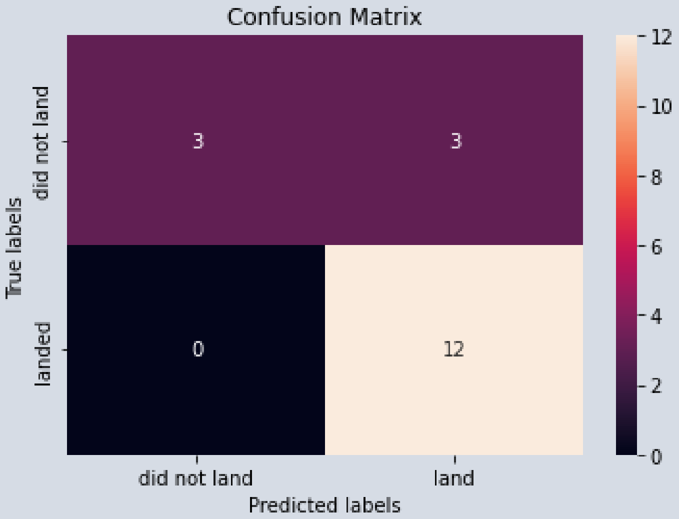
Log. Regression.



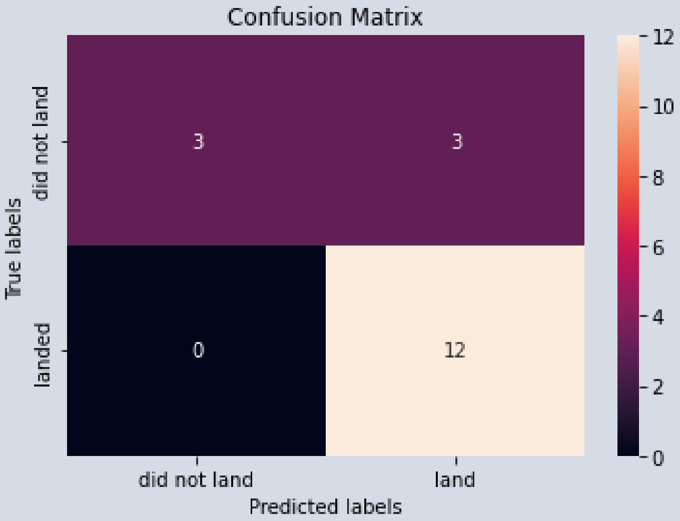
SVM



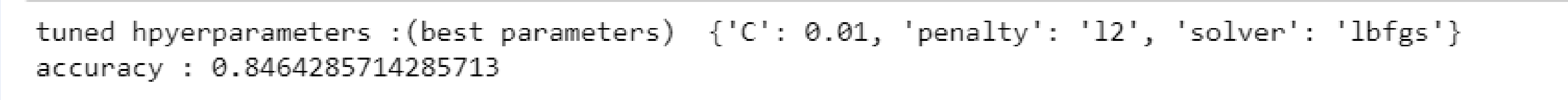
D. Tree



KNN

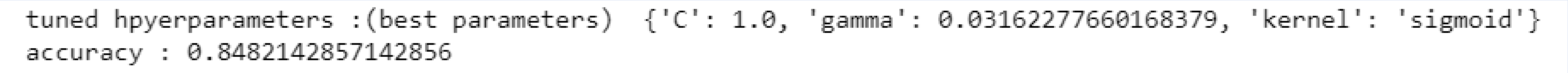


Log. Regression.



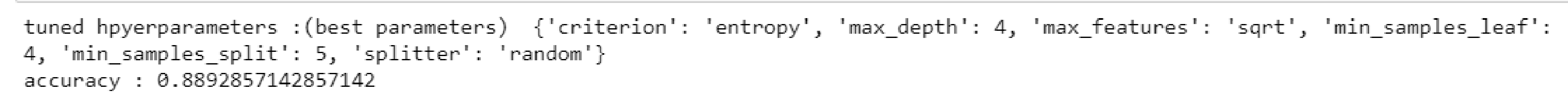
```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

SVM



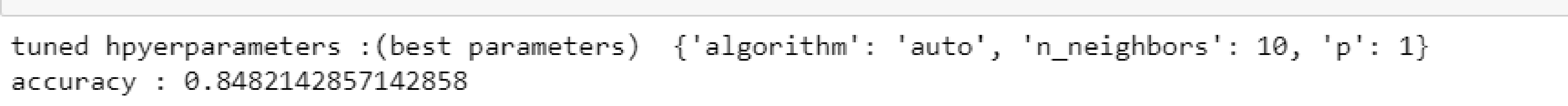
```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

D. Tree

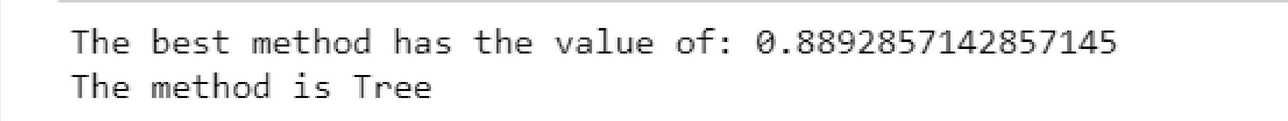


```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.8892857142857142
```

KNN



```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```



```
The best method has the value of: 0.8892857142857145
The method is Tree
```

Conclusion

The biggest number of failed missions is between 2000-6000 kg of pay load mass.

The launch-site with the highest number of successful flights is CCAFS40.

The most succeful missions were on these orbits (ES-LI, GEO, HEO, SSO). However there was not enough data to support it therefore the most successful missions were on ULEO orbit.

The most succesful booster versions between 4000-6000 kg of pay load mass are F9 FT B1 022 and F9 FT 1031 2. The total number of successful missions is 45. There is also the same number of success and failure on ground pads.

The majority of launchsites are 18 km close to city, 1 m close to railway and 0.6 km close to highway.

The best method for the prediction is the Tree with accuracy of 0.89.

The results are discussed on the next slide.

Discussion

- Considering all the findings if the next mission will be on the launch site CCAFS40, on ULEO orbit and with more payload than 6000 kg it with any starter, it will most likely be successful and therefore it will land and stage could be reused. It will also be possible to use less than 6000 kg of payload mass, but the only possible starter versions will be FT B1 022 and F9 FT 1031 2. However, the stage is close to railroads, highways and cities which might make the success of landing lower. On the other hand, most of the missions landed and we could predict whether another stage will be successful or not by using Tree method. If these factors will be kept in mind in future the costs should be minimal and missions should be successful.

Appendix

- Link to Github repository – LINK
- Course - LINK